Domain Adaptation with Clustered Language Models *

J.P. Ueberla
Forum Technology - DRA Malvern, St.Andrews Road
Malvern, Worcestershire, WR14 3PS, UK
email:ueberla@signal.dra.hmg.gb

February 5, 2008

Abstract

In this paper, a method of domain adaptation for clustered language models is developed. It is based on a previously developed clustering algorithm, but with a modified optimisation criterion. The results are shown to be slightly superior to the previously published 'Fillup' method, which can be used to adapt standard n-gram models. However, the improvement both methods give compared to models built from scratch on the adaptation data is quite small (less than 11% relative improvement in word error rate). This suggests that both methods are still unsatisfactory from a practical point of view.

1 Introduction

Current large vocabulary speech recognition systems can achieve good performance on domains for which large quantities (e.g. millions of words) of textual data are available to train a language model. In real world applications, however, this is quite often not the case. The issue of language model domain adaptation is therefore of great practical importance.

One approach to tackle this problem is to try to learn from an analogy to the speaker dependence issue: current systems perform well by training speaker independent models, which can then be adapted with relatively little data from a given speaker (see [8]). Can the same approach be applied to language model adaptation?

In section 2, previous work in this area is reviewed and a rough working definition of domain is given. A method to perform domain adaptation with

*Preprint - To Appear in ICASSP97

clustered language models is then developed (Section 3). Experimental results to evaluate the method are given in Section 4, followed by conclusions in Section 5.

2 Background

In order to make the description of domain adaptation more precise, a definition of domain is needed. One might be tempted to define domain in the sense of semantic topic. However, texts might differ in other aspects (e.g. style), which could still require language model adaptation. A more general definition of domain, more in line with the term sublanguage, is therefore required. According to [10], there are many different definitions of the term, but most of them seem to agree on the following characteristics of a sublanguage:

1. it is part of a natural language
2. it is of a specialised form
3. it behaves like a complete language
4. it is used in special circumstances (e.g. expert communication)
5. it is limited to a particular subject domain

Some of these points seem very useful for the concept of domain (2,4), others less so (1). What properties should an acceptable definition of domain have? The following spring to mind:

- there should be a continuum (e.g. an infinite number) of domains
- each domain may contain an infinite number of elements (e.g. documents/sentences/words)
- for a given element, one should be able to decide whether or not it belongs to a given domain
- all elements of a domain should have a common feature (which defines the domain)

This leads to the following rather wide working definition of domain and hence domain adaptation: A **domain** D is a (often infinite) set of documents such that each document satisfies a property P_D (e.g. 'the document deals with some aspect of law'). Given a sample S_{Back} of domain D_{Back} (background domain) and a sample S_{Adapt} of domain D_{Adapt} (target domain), the problem of language model **domain adaptation** is to produce a language model for D_{Adapt} by using S_{Adapt} and by carrying over some of the information contained in S_{Back}.

Domain adaptation can be divided into static and dynamic domain adaptation, depending on the time scale used to perform adaptation. Dynamic adaptation tries to capture phenomena with a shorter time scale (e.g. topic shifts) and is performed on line, whereas static adaptation can be used to perform a one-time shift from one domain to another and is performed off line. Previous work has shown improvements by using both dynamic adaptation of n-gram models ([7], [5], [2], [9], [6], [13], [3], [4]) and by using static adaptation of n-gram models ([9], [6], [1], [13]). Since the 'Fillup' method presented in [1] gives better performance than linear interpolation, the 'Fillup' method is used as method of comparison for the adaptive clustering, which will be developed in the next section.

3 Adaptive Clustering

The task of a language model is to calculate $p(w_i|c_i)$, the probability of the next word being w_i given the current context c_i. Language models differ in the way this probability is modelled and how the context c_i is defined. A quite general model proposed in [12] makes use of a state mapping function S and a category mapping function G. The idea behind the state mapping $S : c- > s_c = S(c)$ is to assign each of the large number of possible contexts $c \in C$ to one of a smaller number of context-equivalent states. Similarly, the category mapping $G : w- > g_w = G(w)$ assigns each of the large number of possible words $w \in V$ to one of a smaller number of categories (similar to parts of speech). The probability of the next word is then calculated as

$$p(w_i|c_i) = p(G(w_i)|S(c_i)) * p(w_i|G(w_i)). \quad (1)$$

In order to determine S and G automatically, a clustering algorithm as shown in Figure 1 can be used. It is a greedy, hill-climbing algorithm that moves

Algorithm 1: Clustering()

start with initial clustering functions S, G
iterate until some convergence criterion is met
for all $w \in V$ and $c \in C$
for all $g'_w \in G$ and $s'_c \in S$
calculate the difference in the optimisation criterion when w/c is moved from g_w/s_c to g'_w/s'_c
move the w/c to the g'_w/s'_c that results in the biggest improvement in optimisation criterion

End Clustering

Figure 1: The clustering algorithm

elements to the best available choice at any given time. Based on equation 1 and on the leaving-one-out likelihood of the model generating the training data, an optimisation criterion can be derived (see [14] for a detailed description). Let $N(e)$ denote the number of times event e appeared in the training data, let B denote the smoothing parameter used for absolute discounting ([11]), and let n_0, n_1, n_+ denote the number of pairs (s, g) that have appeared zero, one and one or more times in the training data. The resulting optimisation criterion F (as derived in [14]) is

$$\begin{aligned} F &= \sum_{s,g:N(s,g)>1} N(s,g) * log(N(s,g) - 1 - B) \\ &+ n_1 * log(\frac{B * (n_+ - 1)}{(n_0 + 1)}) \\ &- \sum_s N(s) * log(N(s) - 1) - \sum_g N(g) * log(N(g) - 1). \end{aligned} \tag{2}$$

The basic building block in the derivation of equation 2 is the likelihood of one event in the training corpus, as estimated from the training corpus in which this one event has been removed (leaving-one-out likelihood). The main idea behind the adaptive clustering is to use as basic building block the likelihood of one event in S_{Adapt}, as estimated from a linear interpolation of counts from S_{Back} and S_{Adapt} from which this one event has been removed. The motivation for this is that the clustering can thus optimise the perplexity on S_{Adapt}, while having access to a linear combination of counts from S_{Back} and S_{Adapt}.

Let $N_A(e)$ ($N_B(e)$) denote the number of times event e appeared in S_{Adapt} (S_{Back}). Define $N_C(e)$ to be the linear interpolation of the two counts

$$N_C(e) = Round(\lambda * N_A(e) + (1 - \lambda) * N_B(e)) \tag{3}$$

where $Round(x)$ returns the integer nearest to x. The only events that can contribute to the optimisation function are events that occur at least once in S_{Adapt} (because, as explained above, the likelihood of S_{Adapt} is taken as optimisation function). However, their probability is calculated based on the combined counts. Therefore, the smoothing has to apply to the combined counts. Define $n_{bi,0}, n_{bi,1}, n_{bi,+}$ as the number of pairs (s, g) that have a combined count $N_C(s, g)$ of 0, of 1, and larger than 0. In order to introduce absolute discounting for the unigram estimates as well, also define $n_{s,0}, n_{s,1}, n_{s,+}$ as the number of states s that have a combined count $N_C(s)$ of 0, of 1, and larger than 0 (similarly, define $n_{g,0}$ etc. for the unigram estimates involving g). Changing equation 2 according to the basic idea outlined above, this leads to

$$\begin{aligned} &F_{adapt} = \\ &\sum_{s,g:N_A(s,g)>=1, N_C(s,g)>1} N_A(s,g) * log(N_C(s,g) - 1 - B) \\ &+ n_{bi,1} * log(\frac{B * (n_{bi,+} - 1)}{(n_{bi,0} + 1)}) \end{aligned} \tag{4}$$

$$
\begin{aligned}
&- \sum_{s:N_A(s)>=1,N_C(s)>1} N_A(s) * log(N_C(s) - 1 - B) \\
&- \sum_{g:N_A(g)>=1,N_C(g)>1} N_A(g) * log(N_C(g) - 1 - B) \\
&- n_{s,1} * log(\tfrac{B*(n_{s,+}-1)}{(n_{s,0}+1)}) - n_{g,1} * log(\tfrac{B*(n_{g,+}-1)}{(n_{g,0}+1)}).
\end{aligned}
$$

By using the same clustering algorithm as before, but with F_{Adapt} instead of F as optimisation criterion, language model domain adaptation can be performed.

4 Results

In order to test different adaptation methods, two textual samples S_{Back} and S_{Adapt} and acoustic testing data from D_{Adapt} are required. Since the WSJ domain has the associated acoustic data, it is used as D_{Adapt}. As D_{Back}, the patent domain (PAT) was chosen, for which a large sample S_{Adapt} (about 35 million words are used) is also available from the LDC as part of the TIPSTER database.

The recognition system is a state-of-the-art HMM based system (continuous densities, mixtures, triphones). All experiments are based on bigram language models, either clustered (500 clusters) or backoff (singleton bigrams were ignored). The different methods evaluated were

- $Back_{Bo}$: a backoff model built on the background corpus
- $Back_{Cl}$: a clustered model built on the background corpus
- $Adapt_{Cl}$: a clustered model built on the adaptation data
- $Adapt_{Bo}$: a backoff model built on the adaptation data
- $Fillup$: a model built according to the 'Fillup' method presented in [1]
- $ClustAdapt$: a model built with the adaptive clustering presented in the previous section; the initial starting point for the clustering is taken to be the clustering produced by $Back_{Cl}$; one global λ parameter was used and optimised iteratively at the end of each iteration;

For all methods except $Back_{Bo}$ and $Back_{Cl}$, the vocabulary was defined to be all the words that appeared in S_{Adapt}, plus additional words from S_{Back} until 20K words were reached. For $Back_{Bo}$ and $Back_{Cl}$, the vocabulary consisted of the 20K most frequent words in S_{Back}. Because of this difference, the perplexities of $Back_{Bo}$ and $Back_{Cl}$ are not directly comparable to those of the other models. For each method and a given amount of adaptation material, the perplexity of the resulting model was calculated on a held-out section of S_{Adapt} and a recognition run was performed on the acoustic data.

Model	PP	WER (%)
$Back_{Cl}$	955	49.3
$Back_{Bo}$	954	48.4

Table 1: Baseline results

Adapt. words	PP	WER (%)
200	6130	57.0
1000	2740	54.0
5000	1740	47.6
25000	966	39.2
125000	593	33.0

Table 2: Results for $Adapt_{Bo}$

Table 1 gives the results of the two baseline methods, which do not use any of the adaptation material. The high perplexities show that the PAT and WSJ domains are considerably different. The rate of out-of-vocabulary words is about 15%, which is one reason for the very high error rate.

Tables 2, 3, 4 and 5 give the results for the different methods and different amounts of adaptation material.

Comparing Table 3 to Table 2, one can seen that $Adapt_{Cl}$ is more robust than $Adapt_{Bo}$ and it leads to better recognition results for almost the entire range of adaptation material. This is consistent with previous results (see [15]), which showed that clustered models are more robust in terms of perplexity.

Comparing Table 4 to Table 3, one can see that $Fillup$ outperforms $Adapt_{Cl}$ in almost all cases.

By looking at Table 5, one can see that $ClustAdapt$ outperforms $Fillup$ in almost all cases.

Finally, When comparing table 5 to table 3, one can see that the relative improvements in word error rate by using $ClustAdapt$ instead of $Adapt_{Cl}$ are 10.7%, 5.87%, 3.45%, -2.70% and 1.80%.

Adapt. words	PP	WER (%)
200	4170	57.0
1000	2150	51.1
5000	1210	46.4
25000	765	37.0
125000	498	33.4

Table 3: Results for $Adapt_{Cl}$

Adapt. words	PP	WER (%)
200	848	49.9
1000	772	49.6
5000	628	44.9
25000	543	38.2
125000	420	33.0

Table 4: Results for *Fillup*

Adapt. words	PP	WER (%)
200	941	50.9
1000	1040	48.1
5000	821	44.8
25000	801	38.0
125000	623	32.8

Table 5: Results for *ClustAdapt*

5 Conclusion

Compared to the success of some methods for acoustic adaptation, the results obtained here are somewhat disappointing. In particular, they seem to suggest that the improvements from the adaptation techniques compared to starting from scratch on the adaptation data become quite small when several tens of thousands of words are available [1]. One reason for this could be the fact that the acoustic space has an underlying distance metric and thus allows the comparison of two elements. Moreover, one can specify the kind of transformations one would want the adaptation to be able to perform. Both of these points seem more difficult in the case of language model adaptation.

Even though the adaptation method for clustered language models developed in this paper gives slightly better results than the 'Fillup' method, the accuracies obtained with the adaptive clustering and the 'Fillup' method are still very low compared to the about 80% or more the system can achieve with a backoff bigram trained on about 40 million words of the WSJ corpus. Both adaptation methods are therefore still unsatisfactory from a practical point of view.

References

[1] However, it is important to note that this threshold will depend on how dissimilar the two domains are. Moreover, a more fine grained analysis for different amounts of adaptation data would be beneficial, especially since the results for 25,000 words seem to be falling somewhat outside the trend.

[1] Besling and Meier. Language model speaker adaptation. In *European Conference on Speech Communication and Technology*, pages pp.1755–1758. Madrid, Spain, 1995.

[2] S. della Pietra, V. della Pietra, L.R. Mercer, and S. Roukos. Adaptive language modeling using minimum discriminant estimation. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 633–636, 1992.

[3] Marcello Federico. Bayesian estimation methods for n-gram language model adaptation. In *International Conference on Spoken Language Processing*, pages 240–243. Philadelphia, PA, USA, 1996.

[4] R. Iyer and M. Ostendorf. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *International Conference on Spoken Language Processing*, pages 236–240. Philadelphia, PA, USA, 1996.

[5] F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss. A dynamic language model for speech recognition. In *Proceedings of DARPA Speech and Natural Language Workshop*, pages 293–295, February 1991.

[6] Reinhard Kneser and Volker Steinbiss. On the dynamic adaptation of stochastic language models. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 586–589. Minneapolis, Minnesota, USA, 1993.

[7] Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583, 1990.

[8] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer, Speech and Language*, 9:171–185, 1995.

[9] Shoichi Matsunaga, Tomokazu Yamada, and Kiyohiro Shikano. Task adaptation in stochastic language models for continuous speech recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 165–168, 1992.

[10] John McNaught. Introduction to sublanguage. In *Workshop on Sublanguage Grammar and Lexicon Acquisition for Speech and Language Processing*. Speech and Language Technology Club, Jan. 1992.

[11] Hermann Ney and Ute Essen. Estimating 'small' probabilities by leaving-one-out. In *European Conference on Speech Communication and Technology*, pages 2239–2242. Berlin, Germany, 1993.

[12] Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependencies in stochastic language modelling. *Computer, Speech and Language*, 7:101–138, 1993.

[13] Roni Rosenfeld. A maximum entropy approach to adaptive statistical language modelling. *Computer, Speech and Language*, 10:187–228, 1996.

[14] Joerg P. Ueberla. An extended clustering algorithm for statistical language models. Technical Report DRA/CIS(CSE1)/RN94/13, Forum Technology - DRA Malvern, December 1994.

[15] Joerg P. Ueberla. More efficient clustering of n-grams for statistical language modeling. In *European Conference on Speech Communication and Technology*, pages pp.1257–1260, vol.2. Madrid, Spain, 1995.